

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings of claims in the application:

**Listing of Claims:**

- 1                   1.       (currently amended) A system for efficiently performing memory  
2 intensive computations, the system comprising:  
3                   a data cache located in memory of the system, wherein data stored in the data  
4 cache facilitates faster computations on the data stored in the data cache than if the data is stored  
5 in a database, wherein the data cache is coupled to a first set of data stored in a database and a  
6 second set of data stored in memory of the system, wherein the data cache is and configured to  
7 perform a scan operation on at least a portion of the first set of data and an update operation on  
8 the second set of data with changes that have occurred in the first set of data;  
9                   an engine manager coupled to the data cache and configured to instruct the data  
10 cache to perform the scan and update operations; and  
11                   a solver coupled to the data cache and configured to perform one or more  
12 computations on the updated second set of data stored in the data cache, the updated second set  
13 of data including the changes that have occurred in the first set of data.
- 1                   2.       (Previously presented) The system of claim 1, further comprising an  
2 application specific plug-in coupled to the solver and configured to direct the solver to perform  
3 the computations on the second set of data.
- 1                   3.       (Previously presented) The system of claim 1, wherein the system is  
2 configured to update the second set of data with substantially no more than the changes to the  
3 first set of data.
- 1                   4.       (Previously presented) The system of claim 1, wherein the system is  
2 configured to update the second set of data with changes to the first set of data in a near-real-time  
3 fashion.

1                   5.       (previously presented) The system of claim 1, wherein the system is  
2 configured to update the second set of data with substantially no more than the changes to the  
3 first set of data that meet a given condition.

1                   6.       (previously presented) The system of claim 1, wherein the data cache  
2 coupled to a first set of data and a second set of data is coupled in a bidirectional fashion.

1                   7.       (previously presented) The system of claim 1, wherein the first set of data  
2 comprises metadata and application data.

1                   8.       (previously presented) The system of claim 1, wherein the solver  
2 comprises a generic algorithms module.

1                   9.       (previously presented) The system of claim 1, wherein the computations  
2 solve problems encountered in business applications.

1                   10.      (currently amended) A system for efficiently performing memory  
2 intensive computations, the system comprising:

3                   a data cache having a second set of data, the data cache located in memory of the  
4 system, wherein data stored in the data cache facilitates faster computations on the data stored in  
5 the data cache than if the data is stored in a database, wherein the data cache is coupled to a first  
6 set of data stored in the database and, the data cache having a second set of data stored in  
7 memory of the system and configured to perform a scan operation on at least a portion of the first  
8 set of data and an update operation on the second set of data with changes that have occurred in  
9 the first set of data;

10                  an engine manager coupled to the data cache and configured to instruct the data  
11 cache to perform the scan and update operations; and

12                  a solver coupled to the data cache and configured to perform computations on the  
13 updated second set of data stored in the data cache, the updated second set of data including the  
14 changes that have occurred in the first set of data.

1                   11.     (previously presented) The system of claim 10, further comprising an  
2 application specific plug-in coupled to the solver and configured to direct the solver to perform  
3 the computations on the second set of data.

1                   12.     (previously presented) The system of claim 10, wherein the system is  
2 configured to update the second set of data with substantially no more than the changes to the  
3 first set of data.

1                   13.     (previously presented) The system of claim 10, wherein the system is  
2 configured to update the second set of data with the changes to the first set of data in a near-real-  
3 time fashion.

1                   14.     (previously presented) The system of claim 10, wherein the system is  
2 configured to update the second set of data with substantially no more than the changes to the  
3 first set of data that meet a given condition.

1                   15.     (previously presented) The system of claim 10, wherein the data cache  
2 coupled to a first set of data is coupled in a bidirectional fashion.

1                   16.     (previously presented) The system of claim 10, wherein the first set of  
2 data comprises metadata and application data.

1                   17.     (previously presented) The system of claim 10, wherein the solver  
2 comprises a generic algorithms module.

1                   18.     (previously presented) The system of claim 10, wherein the computations  
2 solve problems encountered in business applications.

1                   19.     (currently amended) A system for efficiently performing memory  
2 intensive computations, the system comprising:  
3                   a database comprising a first set of data;  
4                   a plug-in configured to provide application specific functionality; and

5 an in-memory engine located in memory of the system, wherein data stored in the  
6 in-memory engine facilitates faster computations on the data stored in the in-memory engine  
7 than if the data is stored in a database, wherein the in-memory engine is coupled to the database  
8 via a synchronization mechanism and comprising a second set of data stored in memory of the  
9 system, the in-memory engine configured to interface with the plug-in, the in-memory engine  
10 configured to perform computations on the second set of data to derive a first result, the in-  
11 memory engine configured to transfer, when the first set of data changes, these changes to the  
12 second set of data via the synchronization mechanism in order to update the second set of data  
13 and perform computations thereon on the updated second set of data to derive a second result.

1 20. (previously presented) The system of claim 19, wherein the system is  
2 configured such that the second set of data is updated with substantially no more than the  
3 changes to the first set of data.

1 21. (previously presented) The system of claim 19, wherein the system is  
2 configured to update the second set of data with the changes to the first set of data in a near-real-  
3 time fashion.

1 22. (previously presented) The system of claim 19, wherein the system is  
2 configured to update the second set of data with substantially no more than the changes to the  
3 first set of data that meet a given condition.

1 23. (previously presented) The system of claim 19, wherein the in-memory  
2 engine is coupled to the database in a bidirectional fashion.

1 24. (previously presented) The system of claim 19, wherein the database  
2 comprises metadata and application data.

1 25. (previously presented) The system of claim 19, wherein the in-memory  
2 engine comprises a data cache containing the second set of data.

1                   26.     (previously presented) The system of claim 19, wherein the in-memory  
2 engine comprises a generic algorithms module and an in-memory engine manager.

1                   27.     (previously presented) The system of claim 19, wherein the computations  
2 solve problems encountered in business applications.

1                   28.     (currently amended) A method of efficiently performing memory  
2 intensive computations, the method comprising:

3                   performing, using a data cache, a scan operation on at least a portion of a first set  
4 of data;

5                   performing, using the data cache, an update operation on a second set of data with  
6 changes that have occurred in the first set of data, the data cache located in memory, wherein  
7 data stored in the data cache facilitates faster computations on the data stored in the data cache  
8 than if the data is stored in a database; and

9                   performing computations, using a solver, on the updated second set of data stored  
10 in the data cache, the updated second set of data including the changes that have occurred in the  
11 first set of data.

1                   29.     (previously presented) The method of claim 28, wherein the second set of  
2 data is created from substantially all of the first set of data.

1                   30.     (previously presented) (previously presented) The method of claim 28,  
2 wherein the second set of data is created from all of the first set of data.

1                   31.     (previously presented) The method of claim 28, wherein the second set of  
2 data is updated with substantially no more than the changes to the first set of data.

1                   32.     (previously presented) The method of claim 28, wherein the second set of  
2 data is updated with the changes to the first set of data in a near-real-time fashion.

1                   33.     (previously presented) The method of claim 28, wherein the second set of  
2 data is updated with substantially no more than the changes to the first set of data that meet a  
3 given condition.

1                   34.     (previously presented) The method of claim 28, further comprising  
2 updating the first set of data with the second set of data.

1                   35.     (previously presented) The method of claim 28, wherein the computations  
2 solve problems encountered in business applications.

1                   36.     (currently amended) A system for efficiently performing memory  
2 intensive computations, the system comprising:

3                   scanning means for performing, using a data cache, a scan operation on at least a  
4 portion of a first set of data;

5                   update means for performing, using the data cache, an update operation on a  
6 second set of data with changes that have occurred in the first set of data, the data cache located  
7 in memory, wherein data stored in the data cache facilitates faster computations on the data  
8 stored in the data cache than if the data is stored in a database; and

9                   solving means for performing computations, using a solver, on the updated  
10 second set of data stored in the data cache, the updated second set of data including the changes  
11 that have occurred in the first set of data.

1                   37.     (currently amended) A computer-readable medium for efficiently  
2 performing memory intensive computations, the computer-readable medium comprising:

3                   instructions for performing, using a data cache, a scan operation on at least a  
4 portion of a first set of data;

5                   instructions for performing, using the data cache, an update operation on a second  
6 set of data with changes that have occurred in the first set of data, the data cache located in  
7 memory, wherein data stored in the data cache facilitates faster computations on the data stored  
8 in the data cache than if the data is stored in a database; and

9 instructions for performing computations, using a solver, on the second set of data  
10 stored in the data cache, the updated second set of data including the changes that have occurred  
11 in the first set of data.

1 38. (new) A system for efficiently performing memory intensive  
2 computations, the system comprising:

3 a data cache located in memory of the system, wherein data stored in the data  
4 cache facilitates faster computations on the data stored in the data cache than if the data is stored  
5 in a database, wherein the data cache is coupled to a first set of data stored in a database and a  
6 second set of data stored in memory of the system, wherein the data cache is configured to  
7 perform a scan operation on at least a portion of the first set of data and a first update operation  
8 on the second set of data with changes that have occurred in the first set of data;

9 an engine manager coupled to the data cache and configured to instruct the data  
10 cache to perform the first update operation; and

11 a solver coupled to the data cache and configured to perform one or more  
12 computations on the updated second set of data stored in the data cache, the updated second set  
13 of data including the changes that have occurred in the first set of data,

14 wherein the engine manager is configured to determine if the first set of data has  
15 changed since the last update operation, wherein if the first set of data has changed, the engine  
16 manager is configured to perform a second update on the second set of data in the data cache  
17 with the changes to the first set of data since the last update operation,

18 wherein the solver is configured to re-perform the one or more computations on  
19 the updated second set of data stored in the data cache including the changes that have occurred  
20 in the first set of data since the last update operation.